**ABSTRACT**

**ON**

***Interactive MIDI Musical Keyboard: Simplified Music Creation using Python***

*Submitted to*

**DEPARTMENT**

**of**

**CSE(AIML)**

**By**

**Name: Sri Chandra Vardhan Roll No: 2453-21-748-057**

**Name: T Vishnu Vandhan Roll No: 2453-21-748-058**

**Name: A Manidhar Roll No: 2453-21-748038**

**Under the guidance**

**Of**

**Dr B. SUVARNAMUKHI**

**ASSISTANT PROFESSOR**



**NEIL GOGTE INSTITUTE OF TECHNOLOGY**

Kachavanisingaram Village, Hyderabad, Telangana 500058.

**NOVEMBER-2023**

**ABSTRACT**

**INTRODUCTION:**

The objective is to develop a musical keyboard system where each key corresponds to a specific musical note. When a key is pressed, it triggers the corresponding musical sound. The design focuses on creating a user interface that resembles a piano, allowing users to interact with virtual keys and produce music effortlessly. The key emphasis is on simplicity and interactivity, ensuring that users can enjoy playing music without the complexities of raw audio signals or advanced technologies. The project will also incorporate error handling and refinement techniques to provide users with a seamless experience while playing music. The goal is to offer a straightforward yet engaging platform for users to explore and enjoy the world of music through their keyboards.

**EXISTING SYSTEM:**

There are existing systems and software applications based on the concept of mapping keyboard keys to musical notes. These systems are often used in music production, live performances, and educational settings. Some popular examples include:

DAWs (Digital Audio Workstations): Many professional music production software like Ableton Live, FL Studio, and Logic Pro allow users to map computer keyboard keys to musical notes. This feature enables musicians and producers to play virtual instruments using their computer keyboards.

Online Virtual Pianos: Several websites and applications offer virtual pianos where users can play piano notes using their computer keyboards.

These platforms map keys to corresponding piano keys, allowing users to create music online without the need for physical instruments.

Educational Software: Educational software designed for teaching music theory and piano often includes virtual keyboards that can be played using computer keyboards. These tools are valuable for beginners learning to play musical instruments.

Gaming Keyboards with Music Modes: Some gaming keyboards come with customizable keys and modes that allow users to map keys to musical notes or functions. These keyboards can be configured to create music or trigger sound effects during gaming sessions. The limitations of the existing system are as follows:

1. Limited Customizability
2. Complexity for Beginners
3. Lack of Seamless Error Handling
4. Dependency on Specific Software
5. Limited Integration Potential

**PROPOSED SYSTEM:**

The proposed system involves creating a musical keyboard system using Python. It utilizes MIDI libraries like mido or python-rtmidi to generate MIDI signals corresponding to different musical notes. Sound synthesis libraries such as FluidSynth or pygame are employed to produce audio signals directly from MIDI notes without complex processing. Each key on the mechanical keyboard is mapped to a specific MIDI note, triggering the corresponding musical sound. The user interacts with a piano-like interface displayed on the screen using a GUI library like Tkinter. Pressing a key on the physical keyboard lights up the corresponding virtual key on the GUI, providing an interactive experience. When a key is pressed, the mapped MIDI signal is sent to the sound synthesis library, generating the musical note. Error handling is implemented to manage rapid or simultaneous key presses, and user feedback is utilized to refine the user interface and enhance the overall experience. The proposed system is advantageous against the existing systems in many ways:

* Error Handling for Seamless Experience:
* Customizability and Sound Libraries:
* Open-Source and Community Collaboration:
* Integration Potential: